# UNITED STATES PATENT AND TRADEMARK OFFICE

| APPLICATION NO. | FILING DATE | FIRST NAMED INVENTOR | ATTORNEY DOCKET NO. | CONFIRMATION NO. |
|---|---|---|---|---|
| 10/055,645 | 01/22/2002 | Bernard A. Traversat | 5181-82104 | 9627 |

58467          7590          10/24/2008

MHKKG/SUN
P.O. BOX 398
AUSTIN, TX 78767

| EXAMINER |
|---|
| LUU, LE HIEN |

| ART UNIT | PAPER NUMBER |
|---|---|
| 2441 | |

| MAIL DATE | DELIVERY MODE |
|---|---|
| 10/24/2008 | PAPER |

**Please find below and/or attached an Office communication concerning this application or proceeding.**

The time period for reply, if any, is set in the attached communication.

UNITED STATES PATENT AND TRADEMARK OFFICE

Commissioner for Patents
United States Patent and Trademark Office
P.O. Box 1450
Alexandria, VA 22313-1450
www.uspto.gov

# BEFORE THE BOARD OF PATENT APPEALS
# AND INTERFERENCES

Application Number: 10/055,645
Filing Date: January 22, 2002
Appellant(s): TRAVERSAT ET AL.

Robert C. Kowert
For Appellant

## EXAMINER'S ANSWER

This is in response to the appeal brief filed on 07/02/2007 appealing from the Office action mailed on 01/30/2007.

**(1) Real Party in Interest**

A statement identifying by name the real party in interest is contained in the brief.

**(2) Related Appeals and Interferences**

The examiner is not aware of any related appeals, interferences, or judicial proceedings which will directly affect or be directly affected by or have a bearing on the Board's decision in the pending appeal.

**(3) Status of Claims**

The statement of the status of claims contained in the brief is correct.

**(4) Status of Amendments After Final**

The appellant's statement of the status of amendments after final rejection contained in the brief is correct.

**(5) Summary of Claimed Subject Matter**

The summary of claimed subject matter contained in the brief is correct.

**(6) Grounds of Rejection to be Reviewed on Appeal**

The appellant's statement of the grounds of rejection to be reviewed on appeal is correct.

## (7) Claims Appendix

The copy of the appealed claims contained in the Appendix to the brief is correct.

## (8) Evidence Relied Upon

| | | |
|---|---|---|
| 2002/0062375 | Teodosiu et al. | 05/2002 |
| 5,896,503 | Badovinatz et al. | 04/1999 |
| 60252658 | Teodosiu et al. | 11/2000 |

## (9) Grounds of Rejection

The following ground(s) of rejection are applicable to the appealed claims:

1.       The non-statutory double patenting rejection, whether of the obviousness-type or non-obviousness-type, is based on a judicially created doctrine grounded in public policy (a policy reflected in the statute) so as to prevent the unjustified or improper timewise extension of the "right to exclude" granted by a patent and to prevent possible harassment by multiple assignees.  *In re Goodman*, 11 F.3d 1046, 29 USPQ2d 2010 (Fed. Cir. 1993); *In re Longi*, 759 F.2d 887, 225 USPQ 645 (Fed. Cir. 1985) *In re Van Ornum*, 686 F.2d 937, 214 USPQ 761 (CCPA 1982); *In re Vogel*, 422 F.2d 438, 164 USPQ 619 (CCPA 1970); and *In re Thorington*, 418 F.2d 528, 163 USPQ 644 (CCPA 1969).

2.      A timely filed terminal disclaimer in compliance with 37 CFR 1.321(c) may be used to overcome an actual or provisional rejection based on a non-statutory double patenting ground provided the conflicting application or patent is shown to be commonly owned with this application.  See 37 CFR 1.130(b).

3.      Effective January 1, 1994, a registered attorney or agent of record may sign a terminal disclaimer.  A terminal disclaimer signed by the assignee must fully comply with 37 CFR 3.73(b).

4.      Claims 1-40 provisionally rejected under the judicially created doctrine of obviousness-type double patenting as being unpatentable over claims 1-61 of copending Application Number 10/055,649.  Although the conflicting claims are not identical, they are not patentably distinct from each other because the context of the claimed invention is the same as the context of the cited claims of the U.S. patent application.  This is a provisional obviousness-type double patenting rejection because the conflicting claims have not in fact been patented.

5.      The table below shows the similarity of the claimed inventions of application numbers 10/055,645 and 10/055,649.

| 10/055,645 (Claim 1 of the instant application) | 10/055,649 (Claim 18 of copending application) |
|---|---|
| **A peer computing system comprising:** | **A peer computing system, comprising:** |

| a plurality of peer nodes; wherein | plurality of peer nodes operable to couple to a network, wherein the plurality of peer nodes are configure to implement a peer-to-peer environment on the network in accordance with one or more peer-to-peer platform protocols for enabling the plurality of peer nodes to discovery each other, communicate with each other, and cooperate with each other to form peer groups and share network resources in the peer-to-peer environment; one of the plurality of peer nodes operable to maintain two or more mechanisms for accessing a set of peer-to-peer platform resources on the network, wherein the two or more mechanisms are obtainable by devices on the network to enable the devices to participate in the peer-to-peer environment, wherein the two or more mechanisms include: a mechanism for accessing a discovery service for discovering resources in the peer-to-peer environment in accordance with a peer discovery protocol; |
|---|---|
| at least a subset of the peer nodes are configured to participate in a peer discovery protocol to discover other peer nodes; | |
| and wherein at least a subset of the peer nodes are configured to participate in a peer membership protocol for joining or forming a peer group with other peer nodes. | and a mechanism for accessing a membership service for applying for membership in accordance with a peer membership protocol in one or more peer groups each comprising a set of cooperating peer nodes on the network; and a device operable to: couple to the network; obtain the two or more mechanisms from the peer node on the network; and access the set of resources using the two or more mechanisms to participate as a peer node in the peer-to-peer environment. |

6.     The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all

obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this
> title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a
> whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said
> subject matter pertains.  Patentability shall not be negatived by the manner in which the invention was made.

7.     Claims 1-6, 8-18, 21, and 23-40 are rejected under 35 U.S.C. 103(a) as being

unpatentable over Teodosiu et al (US Pub. No. 2002/0062375) and Badovinatz et al

(US Patent 5,896,503).

8.     Claim 1:  Teodosiu teaches a peer computing system comprising:

a plurality of peer nodes (Fig 1, peers 140; page 2, paragraph [0030]); and

wherein at least a subset of the peer nodes are configured to participate in a peer

discovery protocol to discover other peer nodes (pages 2-3, paragraphs [0030-0037]).

However, Teodosiu, fails to teach at least a subset of the peer nodes are

configured to participate in a peer membership protocol for joining or forming a peer

group with other peer nodes.

Badovinatz, teaches a membership protocol for adding nodes to become

members of  a domain in a distributed computing environment which inherently supports

peer-to-peer computing (Figs 1-2, nodes 106s, domains 201A-201D; col. 2 line 30 – col.

3 line 42).  It would be obvious to one of ordinary skill in the computer network art at the

time of the invention to combine the teachings of Teodosiu and Badovinatz to allow peer

nodes to use peer membership protocol for joining or forming a peer group with other

peer nodes because it would   manage membership of a domain of computers of a distributed computing environment.


9.     Claim 2: Teodosiu teaches the peer computing system as claimed, wherein the member peer nodes in said peer group are configured to find and exchange content in said peer group (page 4, paragraph [0045]).


10.     Claim 3. Teodosiu teaches the peer computing system as claimed, wherein said peer group is a collection of cooperating peer nodes that provide a common set of services in the peer computing system (page 2, paragraph [0016]; by definition a peer group is a group of peers communicating with each other and paragraph [0016] teaches accessing the same resource).


11.     Claim 4: Teodosiu teaches the peer computing system as claimed, wherein the common set of services include one or more core services (FIG. 3; wherein the core services are services in the P2P platform).


12.     Claim 5: Teodosiu and Badovinatz teach the peer computing system as claimed, wherein the core services include:

        a discovery service configured for use by member peer nodes in said peer group to discover advertised resources in the peer computing system, wherein the resources include peers and peer groups, and wherein the discovery service uses the discovery

protocol (page 4, paragraph [0053] Teodosiu; wherein the peer node has to advertise its presence and resources for the other peers to discover resources); and

a membership service configured for use by member peer nodes in said peer group to reject or accept group membership applications, wherein the membership service uses the membership protocol (col. 1 lines 40-67 Badovinatz).

13.    Claim 6:  Teodosiu teaches the peer computing system as claimed, wherein one or more peer nodes in said peer group are configured to participate in a peer resolver protocol configured for use in sending search queries from one peer group member to another peer group member (pages 7-8, paragraphs [0094 – 0097]).

14.    Claim 8:  Teodosiu teaches the peer computing system as claimed, wherein one or more peer nodes in said peer group are configured to participate in an endpoint routing protocol for enabling the peer nodes to request peer routing information to reach other peer nodes (page 3, paragraphs [0033 - 0037]; Teodosiu inherently teaches peer nodes can request peer routing information to locate resources).

15.    Claim 9:  Teodosiu teaches the peer computing system as claimed, wherein at least a subset of the peer nodes are configured to participate in a peer information protocol for enabling the peer nodes to learn about other peer nodes' capabilities and status (pages 2-3 and 6, paragraphs [0030 – 0032] and [0073] ).

16.    Claim 10:  Teodosiu teaches the peer computing system as claimed, wherein each of the plurality of peer nodes is further configured to use an advertisement format for describing and publishing advertisements for resources in a peer-to-peer environment (FIG. 3 ref. 380 and paragraph [0073] & [0074]; wherein the passage teaches publishes resources and the resources have to be advertised in order for the other peers or group of peers to learn about the available resources).

17.    Claim 11:  Teodosiu teaches the peer computing system as claimed, wherein the resources include one or more of the peer nodes, peer groups, content, services, applications, pipes, and pipe endpoints, wherein the pipes are communications channels between one or more of the peer nodes, the services, and the applications in the peer-to-peer environment, and wherein the pipe endpoints are network interfaces on the peer nodes that are configured to be bound to the pipes to establish the communications channels (FIGs 1 and 3; pages 5-6, paragraphs [0070 - 0077]).

18.    Claims 12-18, 21, and 23-40 have similar limitations as to claims 1-6 and 8-11; therefore, they are  being rejected under the same rationale as claims 1-6, and 8-11.

19.    Claims 7, 19-20, and 22 would be allowable if rewritten to overcome the rejection under 35 U.S.C. 112 and to include all of the limitations of the base claim and any intervening claims.

**(10) Response to Arguments**

**(I) Applicant's arguments related to double patenting rejection of claims 1-40.**

As to point (I), Examiner stated in the final office action that Claims 1-40 provisionally rejected under the judicially created doctrine of obviousness-type double patenting as being unpatentable over claims 1-61 of copending Application Number 10/055,649. In addition, Examiner provides an analysis of claim 1 of the instant application with claim 18 of copending application 10/055,649 to show the claim 18 contains all limitations of claim 1 of the instant application. In other words, the claim 18 anticipates claim 1 of the instant application. In summary, claims of the instant application therefore are not patently distinct from claims of copending Application Number 10/055,649 and as such are unpatentable over obvious-type double patenting.

| 10/055,645 (Claim 1 of the instant application) | 10/055,649 (Claim 18 of copending application) |
|---|---|
| **A peer computing system comprising:** | **A peer computing system, comprising:** |
| **a plurality of peer nodes; wherein at least a subset of the peer nodes are configured to participate in a peer discovery protocol to discover other peer nodes;** | **plurality of peer nodes operable to couple to a network, wherein the plurality of peer nodes are configure to implement a peer-to-peer environment on the network in accordance with one or more peer-to-peer platform protocols for enabling the plurality of peer nodes to discovery each other, communicate with each other, and cooperate with each other to form peer groups and share network resources in** |

| | the peer-to-peer environment; one of the plurality of peer nodes operable to maintain two or more mechanisms for accessing a set of peer-to-peer platform resources on the network, wherein the two or more mechanisms are obtainable by devices on the network to enable the devices to participate in the peer-to-peer environment, wherein the two or more mechanisms include: a mechanism for accessing a discovery service for discovering resources in the peer-to-peer environment in accordance with a peer discovery protocol; |
|---|---|
| **and wherein at least a subset of the peer nodes are configured to participate in a peer membership protocol for joining or forming a peer group with other peer nodes.** | and **a mechanism for accessing a membership service for applying for membership in accordance with a peer membership protocol in one or more peer groups each comprising a set of cooperating peer nodes on the network**; and a device operable to: couple to the network; obtain the two or more mechanisms from the peer node on the network; and access the set of resources using the two or more mechanisms to participate as a peer node in the peer-to-peer environment. |

**(II) Applicant argued that the 35 U.S.C. 103(a) rejection is improper because the Teodosiu reference is not prior art.**

As to point (II), Both of Teodosiu's Pub. No. 2002/0062375 and provisional application 60/252,658 provide description of locator and tracking service for peer-to-peer resources using Resource Naming Service (RNS). Teodosiu's provisional application 60/252,658 teaches the portions that Examiner relied upon in Teodosiu's

Pub. No. 2002/0062375 to reject applicant claimed invention. Therefore, Teodosiu's

Pub. No. 2002/0062375 is qualified as prior art.


Examiner relies on the following paragraphs in Teodosiu's Pub. No.

2002/0062375 in rejecting applicant claimed invention.

[0016] Furthermore, in the presence of multiple peer copies for the same resource, it is important to be able to select a small set of "best", or "closest", copies for a given request. This ability requires tracking of all equivalent peer locations that have an up-to-date copy of and can serve the cached resource.

[0030] FIG. 1 illustrates one embodiment of the inventive peer-to-peer network. In the illustrated embodiment, peer-to-peer realm 150 (hereafter simply called "realm") includes registrar 110, gate server 120, a number of RNS servers 130, and a number of peers 140. Peers 140 store, or otherwise make available, peer resources (not shown). Registrar 110, RNS servers 130, and gate server 120 together provide a locator and access service for tracking, locating, and accessing the peer resources published in the realm. While one each of the registrar and gate server, and several RNS servers and peers are shown in FIG. 1, the present invention may be practiced with any number of these elements.

[0031] To participate in the realm, each peer 140 first registers with registrar 110. As part of the registration process, registrar 110 assigns each peer an identifier that is unique within realm 150, and also assigns each peer to a particular RNS server 130, hereafter called the "home RNS server" for that peer. The unique identifier for a given peer 140 is used to identify peer resources within realm 150 that are under the control of, or published by, the peer.

[0032] Any number of approaches can be used to register peers 140 with registrar 110. In one embodiment, peers 140 may use a Web-based registration process to obtain and register an identity with registrar 110. Registration may comprise a series of interactions between a peer 140 and registrar 110 to convey a user's identity, encryption keys for secure communications among elements within realm 150, billing information for access various peer resources, downloading and installing software to enable the peer 140 to be compatible with its assigned RNS server, and the like.

[0033] Any number of approaches can also be used by registrar 110 to select a home RNS servers for a particular peer 140. In one embodiment, registrar 110 identifies the type, or version, of software that the peer 140 is using, and generates a list of RNS server 130 that are compatible with that software and that are estimated by registrar 110 to be "close" to the registering peer 140 in terms of network topology. Then, registrar 110 provides this list of candidate RNS servers to the peer 140 so that the peer 140 can test the network paths to each listed RNS server and identify one or more that have the best response times. That is, depending on network topology, the peer 140 is likely to be "closer" to some RNS servers than others, either by physical distance or by the speed of the network medium. In one embodiment, the peer's selection is returned to registrar 110 merely as a suggestion. That is, registrar 110 may or may not assign the peer 140 to the RNS server 130 that the peer selected depending on factors such as the relative network traffic loads among the list of candidate RNS servers. In which case, registrar 110 may only assign the suggested RNS server to be the home RNS server for the peer 140 if the suggested RNS server's relative network traffic load is below a particular level. Otherwise, the

registrar may assign a different RNS server having, for instance, a lowest relative network traffic load among the candidate RNS servers.

[0034] Registrar 110 can also register new RNS servers in realm 150 using many of the same techniques and reassign selected peers 140 to be homed at the new RNS server. That is, in one embodiment, registrar 110 assigns a unique identifier for the RNS server, identifies a set of compatible peers 140, allows the new RNS server to suggest a set of preferred peers, and then selects a set of peers to be homed at the new RNS server based on the suggestion as well as other factors. In alternate embodiments, any number of techniques can be used to register new RNS servers and assign peers. For instance, the new RNS server may simply be allowed to accumulate new peers over time as new peers register with registrar 110.

[0035] Each RNS server 130 tracks the current network location (in terms of IP addresses and IP port numbers) and status (on- or off-line) of all peers assigned to that RNS server, as well as the locations and availability of resources among its assigned peers.

[0036] In general, accessing a resource is a two step process. First, the resource must be located using the locator service. Second, the resource is actually accessed at the location or set of locations returned by the locator service.

[0037] For a peer 140 within realm 150, the first step in accessing a peer resource involves communicating with the peer's assigned home RNS server 130. The home RNS server 130, possibly in cooperation with registrar 110 and another RNS server 130, determines one or more locations within realm 150 where the resource is expected to be available. In one embodiment, the set of locations returned by the home RNS server 130 to the requesting peer 140 may depend on the current network identity (in particular, the current IP address or IP addresses) of peer 140, on the current traffic load on the realm, as well as on other parameters that are known to the RNS servers 130. It is up to the peer 140 to take the second step to actually access the resource at the provided location(s). Once a peer has accessed a resource, it has the option to cache the resource, inform its home RNS server that it is now caching the resource, and make the cached resource available for other peers to access.

[0045] First, the RNS server 130 receives, from a peer 140 or from the gate server 120, a resource request at 210 for the location of a particular resource. The request uniquely identifies a resource and a master publisher of the resource within the realm to which this RNS server belongs. The request can take any number of forms from a messaging protocol specific to this particular locator service to a universally accepted protocol such as HTTP.

[0053] If, in 250, the master publisher is not a local publisher, but the RNS server discovered a publisher record in 240, the RNS server queries the remote RNS server (i.e. the assigned home RNS server for the master publisher) with an identifier of the master publisher and the resource to obtain and cache (by creating a resource record) the status information returned by the remote RNS server for the resource in 270. The remote RNS server may be able to immediately respond to the query based on its own memory records. For instance, the remote RNS server may have records including additional active locations where the resource is expected to be available. If the remote RNS server does not have a record of any active locations for the resource, the remote RNS server may be able to query the master publisher if the master publisher is currently active. Based on the response from the remote RNS server, the local RNS server provides a response in 230. Again, if the resource does not appear to exist, or if the resource is not currently available, the RNS server will respond accordingly. Also, the local RNS server will cache whatever it learns from the remote RNS so that future requests for the resource can be serviced without resorting to the remote RNS. In one embodiment, caching information about resources published by master publishers homed at a remote RNS server can be implemented

on a leasing basis, i.e. the cached information can be assumed to be valid for a specified time after it was obtained from the remote RNS server; additional requests for cached information during that time can then be serviced without contacting the remote RNS server.

[0073] Referring to FIG. 3, peer platform 370 can "publish" peer resources by placing the resources, or a reference to these resources, in publication directory 380. In one embodiment, publishing can be accomplished by the user through an appropriate User Interface provided by platform 370 (not shown in FIG. 3). In one embodiment, publishing can be performed by peer-to-peer applications 345 by calling the appropriate function in the API 340 of platform 370.

[0074] Other devices can browse the contents published by peer 140 by requesting access to the top-level directory or directories published by this peer 140, just like any other resource. Such requests, coming into peer 140 as peer-to-peer traffic 320, may be the result of other users browsing for content, or may be sent at the request of the RNS server, for instance, after a crash as part of a data recovery operation.

[0077] At block 420, the proxy component 360 checks whether this is a request for peer-to-peer content. If it is determined that the request is for a regular network resource published on the Web or on the Intranet, the proxy component 360 just acts as a pass-through for the request, sending the request to the specified Web server or chaining with any existing proxy used by, or specified for, peer 140, as necessary. In one embodiment, platform 370 makes this determination based on the format of the URL provided by peer 140. In another embodiment, platform 370 makes this determination dynamically, as will be explained later under "Dynamic Peer-to-peer Realm Identification." Assuming the request is a regular network resource, for instance having a URL including "www" for a World Wide Web page, the request is forwarded in 425 as regular network traffic 325.

[0094] In one embodiment, a peer-to-peer resource address conforms syntactically to a regular Web URL and cannot be distinguished from a Web URL by any syntactical conventions. Instead, a resource address is dynamically recognized as such when the address is de-referenced. This provides a great deal of flexibility in naming realms. For instance, as discussed above in the embodiment of FIG. 1, external network traffic 125 is received by gate server 120. Gate server 120 can resolve resource addresses and instruct the senders on how to query the resource locator, or gate server 120 can resolve resource addresses and access the resources on behalf of the senders.

[0095] FIG. 6 illustrates one embodiment of resolving a resource address from the perspective of a gate server.

[0096] At 610, the gate server receives a resource address as a regular Web URL.

[0097] At 620, if the requester is a compatible peer device, the gate server instructs the requester in 630 to use its own resource locator service to access the resource. In one embodiment, the gate server may be able to recognize that the requester is a peer device based on a special HTTP header included in the request by the requester. For instance, this can be a "Via" header that specifies the type of the requester as "compatible peer device". Alternately, the requester and the gate server may exchange additional messages to confirm compatibility. If the requester is a compatible peer device, the requester may cache the realm name to avoid sending resource addresses to the gate server in the future.

Teodosiu's provisional application 60/252,685 disclosed that

The Invention described herein is a peer-to-peer locator and tracking service called the Resource Naming Service (abbreviated RNS) that has the following distinguishing characteristics:

- It allows end-user machines to efficiently locate a peer resource, given a peer address for that resource.

- It caches peer files on end-user machines and keeps track of all existing up-to-date cached copies.

- It increases serving bandwidth and peer file availability by directing location requests for a peer file to either the master copy or to one of the cached copies.

- It tracks the availability of end-user machines and their current IP addresses and port numbers. Location requests always resolve to the current coordinates of a machine that can serve the requested peer resource.

- It is scalable to a very large number of peers (hundreds of millions), in a way that is transparent to the peer machines using the service.

- It supports efficient "gates" for compatibility with existing Web and Internet technologies. For instance, an HTTP gate can be provided to make all peer files available on the World Wide Web.

Figure 1 shows the components that make up the RNS system:

- The User Database (UDB). The UDB contains information about the currently registered users (peers) in the system and tracks for each user which RNS Server that user has been assigned to.

- One or more RNS Servers. The RNS Servers maintain information about the published peer content and allow user machines to locate content on their peers. RNS Servers also track the current status and coordinates (IP address and port number) of the peer machines.

- A potentially large number of user machines (peers). The peers communicate with the RNS Servers to identify the location of peer information, and directly to each other to actually transfer that information.

- A Gate Server that provides compatibility with an existing protocol. (For instance, an HTTP Gate Server can make the peer content visible on the World Wide Web.)

Our system relies on a Platform being installed on each of the peer machines, as shown in Figure 2. The Platform turns each peer machine into a server of peer information that has been published on that machine, and into a cache of information that has been accessed from other peers. The Platform includes the following components:

page 3 of Teodosiu's provisional application 60/252,685

- A proxy that intercepts all requests to access peer content originating from the client machine. As part of the compatibility functionality, the proxy can in fact catch *all* requests for external content.

- A server that can serve to other peers content that has either been published on this peer, or that is being cached by the peer. As part of the compatibility functionality, the server can support various protocols, e.g. HTTP.

- An Application Programming Interface (API) that allows peer-to-peer applications to use various services offered by the Platform.

- An RNS interface that is used for communicating with the RNS Server. For instance, this interface is used to relay content location requests to the RNS Server.

Scalability of our system is achieved by partitioning the user space into clusters of user machines that are served by one RNS server. This partitioning is transparent to the end-users. The scheme described herein allows location requests to be resolved efficiently even in the presence of partitioning.


## 3. Detailed description

To use our locator and tracking service, users must first chose a *User Identifier* (UID) and register with the User Database (UDB). This initial registration process can occur via the World Wide Web.

UIDs are used as part of peer addresses to uniquely identify each user in the system. Each user must choose a unique UID; the UDB may place additional restrictions on the UID. For instance, since the UID is a potentially long character string, the system can represent UIDs internally as a fixed-length hash of the string chosen by the user; in this case, not only do the strings have to be unique, but also the hash values. Therefore, certain unique UID strings may be rejected by the UDB since their hash collides with existing values.

At registration time, each user is automatically assigned by the system a *Home RNS Server*. This server is assigned to the user based on some procedure that optimizes the communication patterns in the system (possible criteria could be ping times, or the network topology). The Home RNS Server may change only infrequently, under exceptional circumstances (such as disaster recovery).

The UDB persistently stores various information identifying each user, such as: the UID, the user name, email address, etc. Additionally, the UDB tracks the current home RNS Server for each user. This mapping from UIDs to Home RNS Servers can be queried by any of the RNS Servers in the system as part of the content location process, according to the following protocol:

| RNS$_k$ → UDB | *Home(u)* | *Request home RNS for u* |

page 4 of Teodosiu's provisional application 60/252,685

| UDB → RNS$_k$ | Home(u): RNS$_j$ | Reply with home RNS coordinates |

Since this mapping changes very infrequently, recent home lookups are cached on the RNS Servers to avoid unnecessary traffic to the UDB.

Each RNS Server keeps track of the current status of the peer machines that are homed at that Server. When going online, the Platform establishes contact with the RNS Server using the following protocol:

| P$_k$ → RNS$_i$ | Logon(u,IP,port) | Notify RNS Server that I'm online |
| P$_k$ → RNS$_j$ | Ping(u) | Periodic ping to confirm I'm alive |

As part of the logon protocol, a peer informs its Home RNS Server of its current IP address and port number. This data is stored by the RNS Server and is used when pointing other peers to this machine's current coordinates.

Users can publish files (and possibly other resources, such as devices) on their nodes by creating the files under a special *publication directory*. The Platform tracks all the changes to the publication directory; this is done either by using a file system intercept, or by periodically scanning the file dates under the publication directory in the background. Whenever a file creation, modification, or deletion is detected by the Platform, the latter informs the RNS server using the following protocol:

| P$_k$ → RNS$_j$ | Create(f,v) | File creation |
| P$_k$ → RNS$_i$ | Modify(f,v) | File modification |
| P$_k$ → RNS$_i$ | Delete(f,v) | File deletion |

Since the file name alone is not sufficient to distinguish between different versions of a file, the Platform automatically generates a *version number* v for the file, so that the tuple (f,v) uniquely identifies a particular version of the file. Version information can be generated based on the file creation or modification time, and must be consistent across any operations on the file (such as creation, deletion, and creation of a file with a given name), and across Platform reboots.

The Home RNS Server can at any time request that a peer supply the current version of a given file, as follows:

| RNS$_i$ → P$_k$ | Query(f) | Ask for current version |

| | | |
|---|---|---|
| $P_k \rightarrow RNS_i$ | *Status(f,v)* | *File exists, reply with version* |
| $P_k \rightarrow RNS_i$ | *Status(f,DEL)* | *File has been deleted* |

Version queries are only used infrequently (for infrequently accessed files, for crash recovery, or to resolve race conditions), since the RNS Server normally caches information on the versions of the most recently created or accessed files.

When a user or a peer-to-peer application tries to access a peer resource, the request is intercepted by the Platform. For instance, a user may type a peer HTTP URL into her browser; for this scenario, the Platform acts as an HTTP proxy to the browser. Upon receipt of the URL, the platform applies the URL decision procedure and finds out that this is a peer URL; for a regular World Wide Web the platform would have acted as a pass-through. To resolve a location for the peer content, the Platform communicates with the Home RNS Server:

| | | |
|---|---|---|
| $P_k \rightarrow RNS_i$ | *Locate(f)* | *Locate peer resource* |
| $RNS_i \rightarrow P_k$ | *Location(f,v): IP_i:p_i...* | *Return file location(s)* |
| $P_k \rightarrow RNS_i$ | *Caching(f,v)* | *Inform RNS Server about cache* |

If the RNS Server was able to locate the file, it returns two pieces of information to the requesting peer:

- The current version of the file.

- A list of (IP address, port) pairs of locations that can serve the requested content.

The peer selects one or more locations from the list and contacts these locations directly (by connecting to their instances of the Platform) to effect the actual content transfer. Once the content has been received, the peer has the option of caching it (in addition to passing it to the user or to the requesting application).

Cached files are stored in the Platform cache together with their version numbers. Cached content can be used to serve both local requests, as well as requests from other peers that have been directed to this machine for a cached copy. Local requests can be optimized as follows: if the version of a file returned by the Home RNS Server in the *Location* reply matches that of the cached copy, the request can be satisfied from the local cache, since that copy is up-to-date.

Peers periodically purge their caches of content that was used the least recently (the local file access time information is used to select the content to be purged). Upon purging their cache, peers may notify their Home RNS Server that they are no longer caching that content.

RNS Servers maintain an in-memory mapping from files to current file version and location list. A mapping entry for file f has the following structure:

$$f \rightarrow v, (IP_1:p_1, IP_2:p_2, \ldots)$$

Initially, a file entry will contain only the coordinates of the publishing peer. As the content is being accessed and cached at other locations, the coordinates of the caching locations are accumulated in the list. When an RNS Server fields a *Locate* request for f, it returns a few coordinates from the mapping for f; these coordinates are chosen in a round-robin fashion to spread the load across all peers containing a valid copy of f.

When a peer goes offline (detected at its Home RNS Server by a prolonged absence of *Ping* messages), all entries corresponding to that peer's coordinates are flagged as inactive. When the peer later comes back online, the entries are flagged back as active, and are updated with the peer's latest coordinates (new IP address and port number). An RNS Server never returns inactive coordinates to a *Locate* request.

To minimize memory space requirements at the RNS Server, file names can be represented by a hash computed on the file name string. Collisions are handled by the Platform, which refuses to publish files that collide with already existing ones.

When a Home RNS Server $RNS_1$ is asked to locate a file f corresponding to a user u that is homed at a different RNS Server $RNS_2$, $RNS_1$ needs to contact $RNS_2$ to find out the current location(s) for f. Assuming $RNS_1$ already knows u's home (if it doesn't, it can communicate with the UDB following the protocol described earlier), it interacts with $RNS_2$ as follows:

| $RNS_1 \rightarrow RNS_2$ | *Locate*(f) | *Ask for current coordinates* |
| $RNS_2 \rightarrow RNS_1$ | *Location*(f,v): $IP_1:p_1 \ldots$ | *Reply with version and coordinates* |

The returned coordinates (and version) are stored by $RNS_1$ as a *temporary foreign mapping* for f. This mapping "lives" at $RNS_1$ for a limited time only, and must be renewed after its lifetime has expired. Cached copies of f for peers homed at $RNS_1$ are also accumulated in this mapping, and are used to direct further *Locate* requests for f received by $RNS_1$.

To conclude, compatibility with existing protocols (such as HTTP) is supported as follows. The Gate Server has a dual personality as both a protocol redirector (in the case of HTTP, an HTTP redirector) and an RNS client. When the Gate Server receives a request for a particular piece of peer content, it translates the request address into a peer file name f, corresponding to user u. It then sends a *Locate* request to the Home RNS Server of u to find out the coordinates of a peer that can serve f. Based on these coordinates, it generates a redirect to the original request, pointing it to the peer. This description obviously assumes that the original outside requestor will then be able to talk

directly to the peer to fetch the actual content. Thus, the Platform needs to support the outside protocol (in this case, HTTP).

page 8 of Teodosiu's provisional application 60/252,685


**(III) Applicant's arguments related to limitations in claims 1, 2, 10, and 18.**

Applicant argues that prior art does not teach a peer computing system comprising: a plurality of peer nodes; wherein at least a subset of the peer nodes are configured to participated in a peer discovery protocol to discover other peer nodes.

As to point (III), Teodosiu teaches a peer computing system comprising: a plurality of peer nodes (FIG 1, shows peers 140; page 2, paragraph [0030]); wherein at least a subset of the peer nodes are configured to participated in a peer discovery protocol to discover other peer nodes (pages 2-3, paragraphs [0030-0037]; any peer can register to participate in peer-to-peer realm, and registered peers are assigned unique identifiers. Peer can access to discover peer resources within the realm by communicating with peer's home RNS server).


**(IV) Applicant's arguments related to limitations in claims 1, 2, 10, and 18.**

Applicant argues that prior art does not teach at least a subset of the peer nodes are configured to participated in a peer membership protocol for joining or forming a peer group with other peer nodes.

As to point (IV), Teodosiu teach the invention substantially, but Teodosiu fails to teach at least a subset of the peer nodes are configured to participate in a peer

membership protocol for joining or forming a peer group with other peer nodes. Badovinatz, teaches a membership protocol for adding nodes to become members of a domain in a distributed computing environment which inherently supports peer-to-peer computing (Badovinatz, Figs 1-2, nodes 106s, domains 201A-201D; col. 2 line 30 – col. 3 line 42). It would be obvious to one of ordinary skill in the computer network art at the time of the invention to combine the teachings of Teodosiu and Badovinatz to allow peer nodes to use peer membership protocol for joining or forming a peer group with other peer nodes because it would  manage membership of a domain of computers of a distributed computing environment. The motivation is from Badovinatz's col. 1 lines 5-8.

**(V) Applicant's arguments related to limitations in claims 3 and 4.** Applicant argues that prior art does not teach said peer group is a collection of cooperating peer nodes that provide a common set of services in the peer computing system.

As to point (V), Teodosiu teach multiple peers provide peer copies for the same resources in the realm (page 2, paragraph [0016]).

**(VI) Applicant's arguments related to limitations in claim 5.** Applicant argues that prior art does not teach a discovery service configured for use by member peer nodes in said peer group to discover advertised resources in the peer computing system, wherein the resources include peers and peer groups, and wherein the discovery service uses the discovery protocol and a membership service configured for

use by member peer nodes in said peer group to reject or accept group membership

applications, wherein the membership service uses the membership protocol.

As to point (VI),  Examiner agrees and claim 5 is objected to as being dependent

upon a rejected base claim, but would be allowable if rewritten in independent form

including all of the limitations of the base claim and any intervening claims.


**(VII) Applicant's arguments related to limitations in claims 6 and 21.**

Applicant argues that prior art does not teach one or more peer nodes in said peer

group are configured to participate in a peer resolver protocol configured for use in

sending search queries from one peer group member to another peer group member.

As to point (VII), Teodosiu teaches peer nodes can cache the realm name.

Teodosiu teaches gate server instructs peer nodes to use its own resource locator

service to access the resource in addition to gate sever can resolve resource addresses

(pages 7-8, paragraphs [0094 – 0097]).


**(VIII) Applicant's arguments related to limitations in claims 8 and 23.**

Applicant argues that prior art does not teach one or more peer nodes in said peer

group are configured to participate in an endpoint routing protocol for enabling the peer

nodes to request peer routing information to reach other peer nodes

As to point (VIII), Teodosiu teaches RNS server keeps current network locations

or IP addresses of all peers.  Teodosiu also teaches peers can access and locate IP

addresses to reach other peers (page 3, paragraphs [0033 – 0037]).

**(IX) Applicant's arguments related to limitations in claims 9 and 24.**
Applicant argues that prior art does not teach wherein at least a subset of the peer
nodes are configured to participate in a peer information protocol for enabling the peer
nodes to learn about other peer nodes' capabilities and status.

As to point (IX), Teodosiu teaches peer nodes can identify status of peer
resources within its realm.  Moreover, peer platform can publish peer resources by
placing the resources in publication directory (pages 2-3 and 6, paragraphs [0030 –
0032] and [0073]).


**(X) Applicant's arguments related to limitations in claims 11.**  Applicant
argues that prior art does not teach  the resources include one or more of the peer
nodes, peer groups, content, services, applications, pipes, and pipe endpoints, wherein
the pipes are communications channels between one or more of the peer nodes, the
services, and the applications in the peer-to-peer environment, and wherein the pipe
endpoints are network interfaces on the peer nodes that are configured to be bound to
the pipes to establish the communications channels.

As to point (X), Teodosiu teaches resources in peer-to-peer realm include peers,
servers, content, services, applications, communication channels between any
elements in the peer-to-peer realm.  Teodosiu's peers include network interfaces can be
configured to establish communications channels. Moreover, peer platform can publish
peer resources by placing the resources in publication directory  (FIGs 1 and 3; pages

5-6, paragraphs [0070 - 0077]).


**(XI) Applicant's arguments related to limitations in claim 37.** Applicant argues that prior art does not teach joining the peer group comprises: the peer node sending a peer group membership application message to the peer group, wherein the peer group membership application includes information on the peer node's qualifications for membership in the peer group; the peer group sending a peer group membership acceptance message to the peer node if the peer node qualifies for peer group membership; wherein the peer group membership application message and the peer group membership acceptance message are in a format defined by a peer membership protocol comprised in the peer-to-peer platform.

As to point (XI), Examiner agrees and claim 37 is objected to as being dependent upon a rejected base claim, but would be allowable if rewritten in independent form including all of the limitations of the base claim and any intervening claims.


**(XII) Applicant's arguments related to limitations in claims 12-17, 25-36, 38, and 40.** Applicant presents the same arguments as in claims 1-6, 8-9, and 11.

As to point (XII), Examiner has already discussed how Teodosiu and Badovinatz teach the claimed invention of claims 1-6, 8-9, and 11. Therefore, the same discussion applied for rejecting claims 12-17, 25-36, 38, and 40.

## (11) Related Proceeding(s) Appendix

No decision rendered by a court or the Board is identified by the examiner in the

Related Appeals and Interferences section of this examiner's answer.

For the above reasons, it is believed that the rejections should be sustained.


Respectfully submitted,

/Le Luu/

Primary Examiner, Art Unit 2441


Conferees:

*/Stephen S. Hong/*

*Supervisory Patent Examiner, Art Unit 2178*


/Dave Wiley/
Supervisory Patent Examiner, Art Unit 2174


Meyertons, Hood, Kivlin, Kowert & Goetzel, P.C.
P.O. Box 398
Austin, TX 78767 – 0398
(512) 853-8850